THE ASSISTANT COMMISSIONER OF PATENTS
Washington, D.C. 20231

DOCKET NUMBER: AT9-99-367
September 23, 1999

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventor(s): Viktors Berstis

For: **METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR KEEPING FILES CURRENT**

Enclosed are:

__X__   Patent Specification and Declaration

__X__   _Three_ sheets of drawing(s).

__X__   An assignment of the invention to International Business Machines Corporation
(includes Recordation Form Cover Sheet).

_____   A certified copy of a _____ application.

_____   Information Disclosure Statement, PTO 1449 and copies of references.

The filing fee has been calculated as shown below:

For	Number Filed	Number Extra	Rate	Fee
Basic Fee				$760.00
Total Claims	33 - 20	13	x 18 =	$234.00
Indep. Claims	3 - 3		x 78 =	$
MULTIPLE DEPENDENT CLAIM PRESENTED			x 260 =	$
			TOTAL	$894.00

__X__   Please charge IBM Corporation Deposit Account No. _09-0447_ in the amount of _$894.00_.
A duplicate copy of this sheet is enclosed.

__X__   The Commissioner is hereby authorized to charge payment of the following fees
associated with this communication or credit any overpayment to IBM Corporation
Deposit Account _09-0447_.

__X__   Any additional filing fees required under 37 CFR §1.16.

__X__   Any patent application processing fees under 37 CFR §1.17.

Respectfully submitted,

By _Richard McCain_

Richard N. McCain
Registration No. 43,785
FELSMAN, BRADLEY, VADEN,
    GUNTER & DILLON, LLP
Suite 350, Lakewood on the Park
7600B North Capital of Texas Highway
Austin, Texas 78731
Telephone (512) 343-6116

METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT

FOR KEEPING FILES CURRENT

BACKGROUND OF THE INVENTION

5   1.   Field of the invention:

The present invention relates generally to data processing and in particular to file management. Still more particularly, the present invention relates to a method, system and computer program product for keeping
10   files current.

2.   Description of related art:

"Network computing" in a literal sense means an environment wherein a number of computers and/or peripheral devices are connected together by a
15   communication medium (whether it be a wired or wireless medium). Additionally, the term "network" also means a communication network for transmitting data between devices that are connected to the network, such as computers, printers, storage devices and the like. There
20   are diverse forms of networks that range from a local area type, such as a local area network (LAN), to a wide area type such as a public switched telephone network (PSTN) and further to the "Internet" that has grown to a large collection of global networks as a result of
25   interconnecting respective servers.

A LAN is a smallest unit of a network, which is autonomously operated/managed by an independent organization, such as a college or research institution to cover a relatively narrow area, e.g., a single campus
30   or the like. Supported with the price reduction of communication equipment reflecting the evolution of

semiconductor technologies and the enhanced functions of
communication software, LANs have been primarily used in
areas, such as in the research/development arenas, for
the purpose of sharing computer resources,

5   sharing/distribution of information and the like.

           Wide area networks (WANs), on the other hand, are,
in a simplistic sense, a larger collection of LANs
wherein the servers that service each individual LAN are
interconnected to create a larger network environment.

10   Thus the services, e.g., sharing/distribution of
information, are made available on a much larger global
arena.

           The emergence of wide area network systems, such as
the Internet, has increased the accessibility of

15   information. Connected users within these network
systems have access to useful information that is made
publicly available from locations, or sites, such as
research facilities and libraries. These publicly
available information are typically downloaded by a user,

20   e.g., in the form of ZIP and PDF files, that are then
saved on the user's memory storage devices, e.g., hard
disk drive and writeable CDROM.

           The files that have been downloaded may typically
reside in the user's memory devices for extended periods

25   of time prior to the information contained in those files
being accessed by the user. During this extended period
of time, which may be months or years, the information
may become outdated or updates may exist that correct
errors that have been identified in the version of the

30   file that was downloaded. Furthermore, with the passage
of time, the user may not remember the location from
where the file originated and determining that location
may be a difficult, if not impossible, task if the user

decides to check for a updated or newer version.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention
to provide a method, system and computer program product
for keeping files within a data processing system
current.

To achieve the foregoing object, and in accordance
with the invention as embodied and broadly described
herein, a method, system and computer program product are
disclosed for keeping files current for use in a computer
system coupled to a network. The method includes: (1)
evaluating a downloaded file from a source within the
network to determine if a source identifier is present in
the downloaded file, (2) checking the source periodically
utilizing the source identifier to determine if a newer
version of the downloaded file exists and (3) replacing,
in response to the presence of a newer version of the
downloaded file, the downloaded file with the newer
version. The method further includes attaching, in
response to the source identifier not being present, a
source descriptor to the downloaded file.

In one embodiment of the present invention, the step
of replacing the downloaded file includes the steps of
(1) providing an indication to a user that the newer
version of the file exists, (2) prompting the user to
replace the downloaded file with the newer version and
(3) replacing, in response to the user requesting the
newer version, the downloaded file with the newer
version.

In another embodiment of the present invention, the
source identifier is located in the extended attribute of
the downloaded file. It should be noted, however, that
the location of the source identifier may vary depending

on the type of file format or operating system employed.

In yet another embodiment of the present invention, the downloaded file is a PDF file. Alternatively, in another advantageous embodiment, the downloaded file is a ZIP file. It should be readily apparent to those skilled in the art that the present invention may be advantageously practiced with other file format methodologies.

In another embodiment of the present invention, the step of checking the source periodically includes defining a time interval. In one advantageous embodiment, the time interval is user defined. Alternatively, the step of checking the source may be accomplished whenever the downloaded file is opened or "on-demand" by a user.

In one embodiment of the present invention, the network is a packet network. Of course, the present invention may also be advantageously practiced in other network environments such as local area networks (LANs) and wide area networks (WANs). The present invention does not contemplate limiting its use to any one particular network environment.

The foregoing description has outlined, rather broadly, preferred and alternative features of the present invention so that those skilled in the art may better understand the detailed description of the invention that follows. Additional features of the invention will be described hereinafter that form the subject matter of the claims of the invention. Those skilled in the art should appreciate that they can readily use the disclosed conception and specific embodiment as a basis for designing or modifying other

structures for carrying out the same purposes of the
present invention. Those skilled in the art should also
realize that such equivalent constructions do not depart
from the spirit and scope of the invention in its

5   broadest form.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present
invention, reference is now made to the following
descriptions taken in conjunction with the accompanying
5   drawings, in which:

FIGURE 1 illustrates an exemplary network system
that provides a suitable environment for the practice of
the present invention;

FIGURE 2 illustrates an embodiment of a controller
10   employing a file updating system constructed utilizing
the principles disclosed by the present invention; and

FIGURE 3 illustrates a high level logic flow diagram
of an embodiment of a file updating process utilizing the
principles disclosed by the present invention.

DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENTS

With reference now to the figures, and in particular, with reference to **FIGURE 1**, there is depicted an exemplary network system **100** that provides a suitable environment for the practice of the present invention. Network system **100** includes a computer system **110**, such as a personal computer (PC), that is coupled to first and second sites **120**, **130**, respectively, via a packet network **140**, e.g. the Internet. It should be noted that the present invention may also be advantageously practiced in other network environments, such as a local area network (LAN). First and second sites **120**, **130** are generally sites that provide information to users, such as libraries and research facilities, that are connected to network system **100**. First and second sites **120**, **130** typically provide services, which may be free, i.e., no monetary charges are required to access the site services, that include application programs, such as Acrobat reader from Adobe. These "free" programs are generally available in a file that a user, such as computer system **110**, would download through packet network **140** to a memory device (not shown), such as a hard disk or a writeable CDROM, coupled to computer **110**.

As discussed previously, the files that have been downloaded may typically reside in the user's memory devices for extended periods of time prior to the information contained in those files being accessed by the user. During this extended period of time, which may be months or years, the information may become out dated or updates may exist that correct errors that have been identified in the version of the file that was downloaded. Furthermore, with the passage of time, the user may not remember the location from where the file originated and determining that location may be a

difficult, if not impossible, task if the user decides to check for a updated or newer version.

Referring now to **FIGURE 2**, there is illustrated an embodiment of a controller **200** employing a file updating system constructed utilizing the principles disclosed by the present invention. Controller **200** (analogous to computer system **100** illustrated in **FIGURE 1**), in an advantageous embodiment, is a personal computer manufactured by IBM Corporation of Armonk, N.Y. It should also be readily apparent to those skilled in the art, however, that alternative computer system architectures may be employed. Generally, controller **210**, embodied in a PC computer, comprises a bus **215** for communicating information, a processor **220** coupled to bus **215** for processing information, a random access memory (not shown) coupled to bus **215** for storing information and instructions for processor **220**, a read-only memory (not shown) coupled to bus **215** for storing static information and instructions for processor **220**, a display device **250** coupled to bus **215** for displaying information for a computer user, an input device (not shown) coupled to bus **215** for communicating information and command selections to processor **220** and a data storage device (not shown), such as a magnetic disk and associated disk drive, coupled to bus **215** for storing information and instructions.

Processor **220** may be any of a wide variety of general purpose processors or microprocessors, such as the i486™ or Pentium™ brand microprocessor manufactured by Intel Corporation of Santa Clara, California. However, it should be apparent to those skilled in the art that other varieties of processors may be utilized in a computer system. Display device **250** may be a liquid crystal device, cathode ray tube (CRT), or other suitable

display device. The data storage device may be a
conventional hard disk drive, floppy disk drive, or other
magnetic or optical data storage device for reading and
writing information stored on a hard disk drive, floppy
5   disk drive, or other magnetic or optical data storage
medium.

In general, processor **220** retrieves processing
instructions and data from a data storage medium using
the data storage device and downloads this information
10   into random access memory for execution. Thereafter,
processor **220** then executes an instruction stream from
random access memory or read only memory. Command
selections and information input at the input device are
used to direct the flow of instructions executed by
15   processor **220**. The results of this processing execution
are then displayed on display device **250**.

Controller **210** further includes an update manager
230 that is coupled to processor **220**. Update manager
230, in an advantageous embodiment, is embodied as a set
20   of computer executable instructions stored on a computer
readable medium, such as the hard disk. It should be
readily apparent, however, to those skilled in the art
that update manager **230** may also be implemented in
hardware, firmware, software and any combination thereof.
25   The present invention does not contemplate limiting its
practice to any particular form of implementation.

Referring now to **FIGURE 3**, with continuing reference
to **FIGUREs 1 and 2**, depicted is a high level logic flow
diagram of an embodiment of a file updating process **300**
30   utilizing the principles disclosed by the present
invention. Process **300** begins, as depicted in step **310**,
when the process is queued for execution. Next, as
illustrated in step **320**, controller **210** selects and

downloads a file from a source site, e.g., first or
second sites **120**, **130**. In an advantageous embodiment,
the downloaded file may be a PDF file. Alternatively, in
another advantageous embodiment, the downloaded file is a

5   ZIP file. It should be noted, however, that the practice
of the present invention is not limited to any particular
type of file format methodology.

Following the downloading of the selected file,
update manager **230** evaluates the downloaded file to

10   determine if the file has a source identifier associated
with the site from which it was obtained, as depicted in
decisional step **330**. The evaluation is accomplished,
e.g., by looking at the extended attributes or directory
in the downloaded file, to see if an identifier, such as

15   an uniform resource locator (URL), associated with the
site is present. If it is determined that there is no
source identifier associated with the downloaded file,
update manager attaches a source descriptor to the
downloaded file, as illustrated in step 340. The

20   attachment or "tagging" the source descriptor may be
accomplished, e.g., by adding a new comment entry in a
ZIP file. Generally, most relevant file formats have
room for additional comment text or other attribute
string. This attribute string, i.e., "source

25   identifier," is added to the file to identify the source
location of the file to which it is attached or in which
it appears. It should be noted that certain operating
systems, such as OS/2, support extended attributes that
are associated with a file. Therefore, the source

30   identifier may be stored as an extended attribute and
does not need to be inserted inside the file. The source
identifier, in an advantageous embodiment, may contain
the following:

(1) A signature string that is unlikely to appear in

any other portion of the file. This signature string is used to find the source identifier within the file.

(2) A URL or other locator string that identifies the location from which the file (its newest version) can be retrieved.

(3) A date/time and version number corresponding to the file.

(4) A checksum string covering the prior entries to make it less likely that random data content would be mistaken for a signature string.

When the source identifier is located within the file, it should be also located as far towards the end of the file as possible, so that the last signature string in the file is the one that is a part of the source identifier. In the event that an uncompressed archive file, such as a ZIP file, contains other ZIP files with their own source identifiers, locating the source identifier at the end of the file would prevent the present invention from incorrectly using an earlier embedded source identifier in the file. It should be readily apparent to those skilled in the art that the preferred location of the source identifier is different for different file types and depends on the methodology employed by the file to contain comment strings.

Alternatively, in another advantageous embodiment, an entry may be entered in a specially coded file registry associated within controller **210** that records, at a minimum, the name of the downloaded files and a source descriptor identifying the originating source, such as an URL, from where the downloaded file was obtained. Other entries in this file registry may

include a time stamp of when the downloaded file was
retrieved from the source site. In the case where
controller **210** is running an OS/2 operating system, the
source descriptor information may be stored in the
extended attributes of the downloaded file. It should
readily apparent to those skilled in the art that the
location where the source descriptor is stored is
dependent on the file format methodology employed by
controller **210**, e.g., for a PDF file, the source
descriptor information may not be added to the file in a
manner that may disable older versions of Acrobat
viewers. In the case of a PDF file or with the OS/2
operating system discussed above, in an advantageous
embodiment, the source descriptor could be stored within
the file by replacing a text of a comment or any other
embedded string in the document file with a specially
coded string that has an unique digital signature.

Following the attachment of the source descriptor,
process **300** enters a "dormant" or waiting period, as
depicted in step **345**, until such time as when the
downloaded file is opened by a user or, in another
advantageous embodiment, at a predetermined time
interval. The time interval is typically set by the user
and may be programmed to be as short as daily or longer
as every six months. Alternatively, in another
embodiment, a triggering event may be a "on-demand"
request by the user to update the file. Similarly, if it
is determined in step **330** that the downloaded file has a
source identifier included in it, process **300** proceeds to
wait until the downloaded file is opened by the user or
at the predetermined time interval.

In the event that the downloaded file is opened by
the user, or in another alternative embodiment, at the
expiration of the predetermined time interval, update

manager **230** retrieves the source identifier from the
downloaded file and proceeds to check the file's source
site to determine if a newer version of the file is
present, as depicted in decisional step **350**.

5   If there is a newer version of the file present, the
update manager **210** proceeds to provide an indication to
the user that a newer version of the file is available.
The user may be prompted with a message, such as "Updated
version of file available, would you like to replace
10   existing file Y/N" displayed on display **250**. If the user
responses with an affirmative Y, update manager **230**
replaces the "older" file with its newer or updated
version as illustrated in step **260**. Alternatively, in
another embodiment, replacing the older file involves
15   renaming the older file. With this approach, the older
version of the file is still available along with the
newer, i.e., most current, file version. After update
manager has replace the file with its newer version or if
the user had decided that the newer version is not
20   desired, process **300** does nothing and returns to its
dormant state, i.e., step **345**, to wait for the next
triggering event, e.g., when the file is opened again,
on-demand by the user or at the end of the next time
interval.

25   The present invention provides for the automatic
updating of files that have been downloaded from a source
site that is coupled to a user's system without the user
having to remember where the file was obtained from.
Consequently, with an attached source identifier, any
30   downloaded file can be traced to its originating source
site. Furthermore, mirror sites can perform automatic
file updates based on the file content rather than using
a separate directory of file locations.

It should be noted that although the present
invention has been described in the context of a computer
system, those skilled in the art will readily appreciate
that the present invention is also capable of being

5   distributed as a computer program product in a variety of
forms; the present invention does not contemplate
limiting its practice to any particular type of signal-
bearing media, i.e., computer readable medium, utilized
to actually carry out the distribution. Examples of

10   signal-bearing media includes recordable type media, such
as floppy disks and hard disk drives, and transmission
type media such as digital and analog communication
links.

In a preferred embodiment, the present invention is

15   implemented in a computer system programmed to execute
the method described herein. Accordingly, in an
advantageous embodiment, sets of instructions for
executing the method disclosed herein are resident in RAM
of one or more of computer systems configured generally

20   as described hereinabove. Until required by the computer
system, the set of instructions may be stored as computer
program product in another computer memory, e.g., a disk
drive. In another advantageous embodiment, the computer
program product may also be stored at another computer

25   and transmitted to a user's computer system by an
internal or external communication network, e.g., LAN or
WAN, respectively.

The present invention may be embodied in other
specific forms without departing from its spirit or

30   essential characteristics. The described embodiments are
to be considered in all respects as illustrative and not
restrictive. The scope of the invention is, therefore,
indicated by the appended claims rather than by the
foregoing description. All changes which come within the

meaning and range of equivalency of the claims are to be
embraced within their scope.

CLAIMS:

What is claimed is:

1   1.   A method for keeping files current for use in a
2   computer system coupled to a network, comprising the
3   steps of:

4   evaluating a downloaded file from a source within
5   said network to determine if a source identifier is
6   present in said downloaded file;

7   checking said source periodically utilizing said
8   source identifier to determine if a newer version of said
9   downloaded file exists; and

10   replacing, in response to the presence of said newer
11   version of said downloaded file, said downloaded file
12   with said newer version.

1   2.   The method as recited in claim 1 wherein said step
2   of evaluating further includes the step of attaching, in
3   response to said source identifier not present, a source
4   descriptor to said downloaded file.

1   3.   The method as recited in Claim 1 wherein said step
2   of replacing said downloaded file includes the steps of:

3   providing an indication to a user that said newer
4   version of said file exists;

5   prompting said user to replace said downloaded file
6   with said newer version; and

7   replacing, in response to said user requesting said
8   newer version, said downloaded file with said newer
9   version.

1
2
3

4. The method as recited in Claim 1 wherein said source identifier is located in the extended attribute of said downloaded file.

1
2
3

5. The method as recited in Claim 1 wherein said downloaded file is selected from the group consisting of PDF and ZIP files.

1
2

6. The method as recited in Claim 1 wherein said source identifier is an uniform resource locator (URL).

1
2
3

7. The method as recited in Claim 1 wherein said step of checking said source periodically includes defining a time interval.

1
2

8. The method as recited in Claim 7 wherein said time interval is user defined.

1
2
3

9. The method as recited in Claim 1 wherein said step of checking said source is whenever said downloaded file is opened.

1
2

10. The method as recited in Claim 1 wherein said step of checking said source is initiated by a user.

1
2

11. The method as recited in Claim 1 wherein said network is a packet network.

1   12.   A computer system for use in a network environment,
2   comprising:

3   a processor;

4   an update manager coupled to said processor,
5   including:

6   means for evaluating a downloaded file from a
7   source within said network to determine if a source
8   identifier is present in said downloaded file;

9   means for checking said source periodically
10   utilizing said source identifier to determine if a
11   newer version of said downloaded file exists; and

12   means for replacing, in response to the
13   presence of a newer version of said downloaded file, said
14   downloaded file with said newer version.

1   13.   The computer system as recited in Claim 12 wherein
2   said means for evaluating further includes means for
3   attaching, in response to said source identifier not
4   present, a source descriptor to said downloaded file.

1   14. The computer system as recited in Claim 12 wherein
2   said means for replacing said downloaded file includes:

3   means for providing an indication to a user that
4   said newer version of said file exists;

5   means for prompting said user to replace said
6   downloaded file with said newer version; and

7   means for replacing, in response to said user
8   requesting said newer version, said downloaded file with
9   said newer version.

1   15. The computer system as recited in Claim 12 wherein
2   said source identifier is located in the extended
3   attribute of said downloaded file.

1   16. The computer system as recited in Claim 12 wherein
2   said downloaded file is selected from the group
3   consisting of PDF and ZIP files.

1   17. The computer system as recited in Claim 12 wherein
2   said source identifier is an uniform resource locator
3   (URL).

1   18. The computer system as recited in Claim 12 wherein
2   said means for checking said source periodically includes
3   defining a time interval.

1   19. The computer system as recited in Claim 18 wherein
2   said time interval is user defined.

1   20. The computer system as recited in Claim 12 wherein
2   said means for checking said source is whenever said
3   downloaded file is opened.

1   21.   The computer system as recited in Claim 12 wherein
2   said means for checking said source is initiated by a
3   user.

1   22.   The computer system as recited in Claim 12 wherein
2   said network is a packet network.

1   23.   A computer program product comprising:

2   a computer-readable medium having stored thereon
3   computer executable instructions for implementing a
4   method for keeping files current for use in a computer
5   system coupled to a network, said computer executable
6   instructions when executed, perform the steps of:

7   evaluating a downloaded file from a source
8   within said network to determine if a source identifier
9   is present in said downloaded file;

10   checking said source periodically utilizing
11 said source identifier to determine if a newer version of
12 said downloaded file exists; and

13   replacing, in response to the presence of said
14 newer version of said downloaded file, said downloaded
15 file with said newer version.

1   24.   The computer program product as recited in Claim 23
2   wherein said step of evaluating further includes the step
3   of attaching, in response to said source identifier not
4   present, a source descriptor to said downloaded file.

1   25. The computer program product as recited in Claim 23
2   wherein said step of replacing said downloaded file
3   includes the steps of:

4   providing an indication to a user that said newer
5   version of said file exists;

6   prompting said user to replace said downloaded file
7   with said newer version; and

8   replacing, in response to said user requesting said
9   newer version, said downloaded file with said newer
10 version.

1   26. The computer program product as recited in Claim 23
2   wherein said source identifier is located in the extended
3   attribute of said downloaded file.

1   27. The computer program product as recited in Claim 23
2   wherein said downloaded file is selected from the group
3   consisting of PDF and ZIP files.

1   28. The computer program product as recited in Claim 23
2   wherein said source identifier is an uniform resource
3   locator (URL).

1   29. The computer program product as recited in Claim 23
2   wherein said step of checking said source periodically
3   includes defining a time interval.

1   30. The computer program product as recited in Claim 29
2   wherein said time interval is user defined.

1   31. The computer program product as recited in claim 23
2   wherein said step of checking said URL is whenever said
3   downloaded file is opened.

1   32.   The computer program product as recited in Claim 23
2   wherein said step of checking is initiated by a user.

1   33.   The computer program product as recited in Claim 23
2   wherein said network is a packet network.

ABSTRACT OF THE DISCLOSURE

METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT
FOR KEEPING FILES CURRENT

1   A method, system and computer program product for

2   keeping files current for use in a computer system

3   coupled to a network. The method includes: (1)

4   evaluating a downloaded file from a source within the

5   network to determine if a source identifier is present in

6   the downloaded file, (2) checking the source periodically

7   using the source identifier to determine if a newer

8   version of the downloaded file exists and (3) replacing,

9   in response to the presence of a newer version of the

10   downloaded file, the downloaded file with the newer

11   version. The method further includes attaching, in

12   response to the source identifier not being present, a
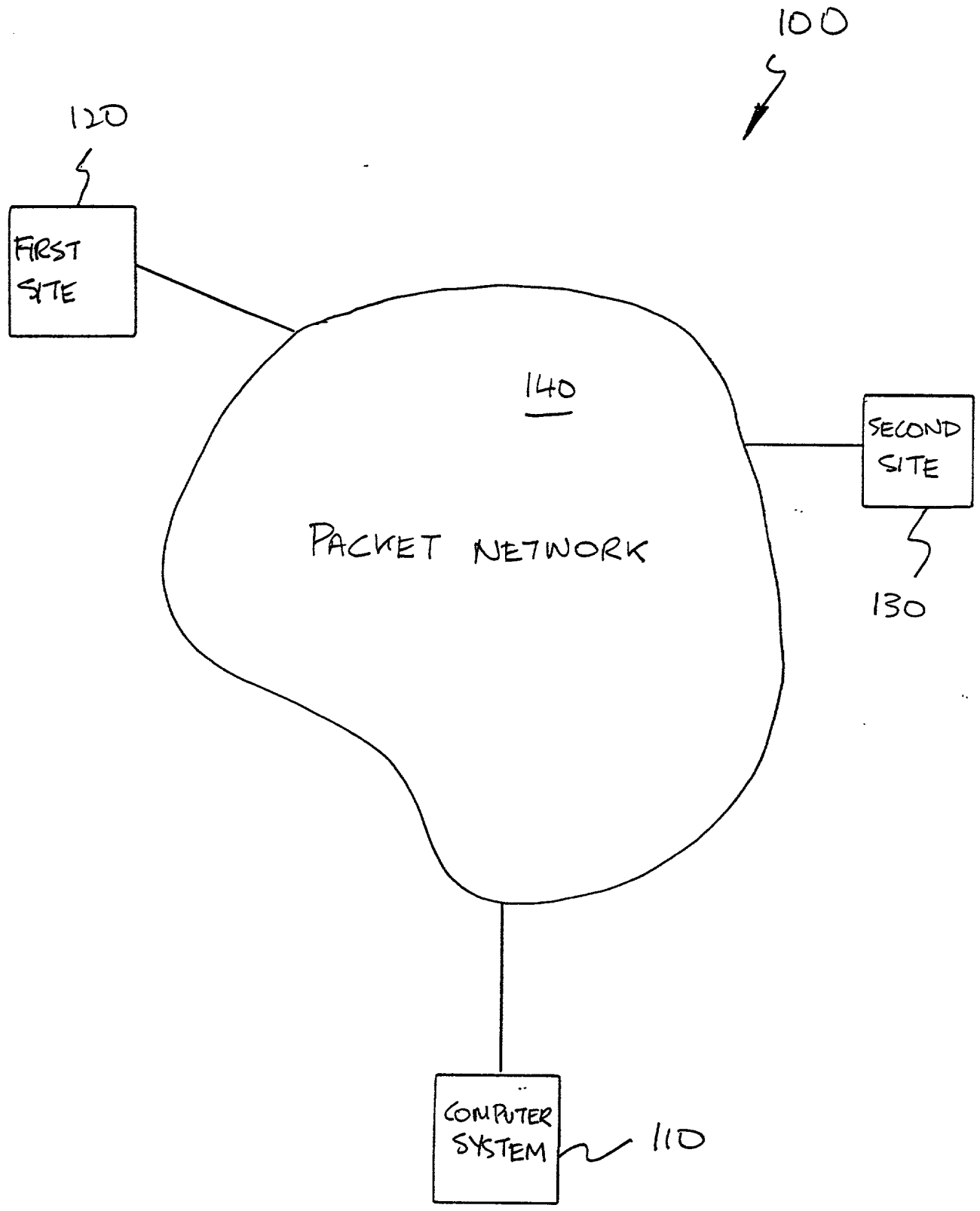
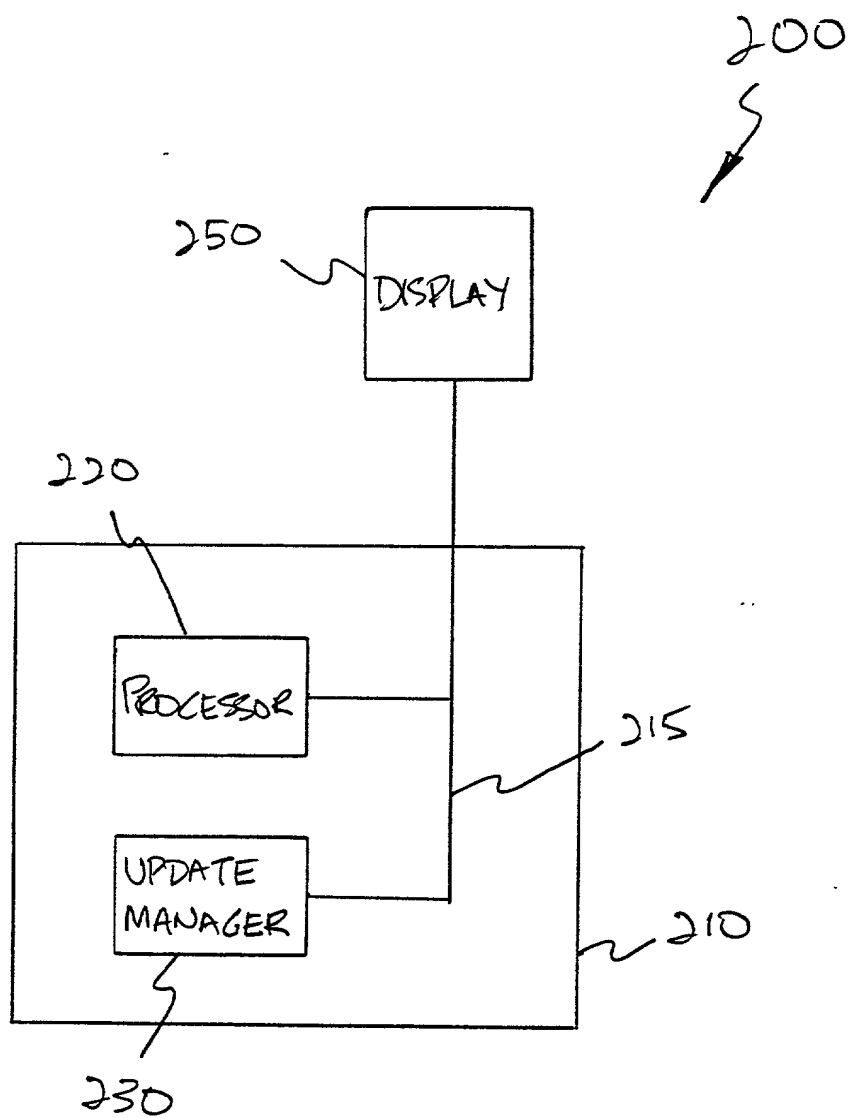13   source descriptor to the downloaded file.

100

120

FIRST
SITE

140

PACKET NETWORK

SECOND
SITE

130

COMPUTER
SYSTEM

110

FIGURE 1 (PRIOR ART)

FIGURE 2

300

BEGIN 310

DOWNLOAD A FILE FROM
A SOURCE SITE 320

330

Y

DOES FILE HAVE
A SOURCE IDENTIFIER?

N

ATTACH A SOURCE
DESCRIPTOR TO FILE 340

WAIT 345

350

N

IS THERE A
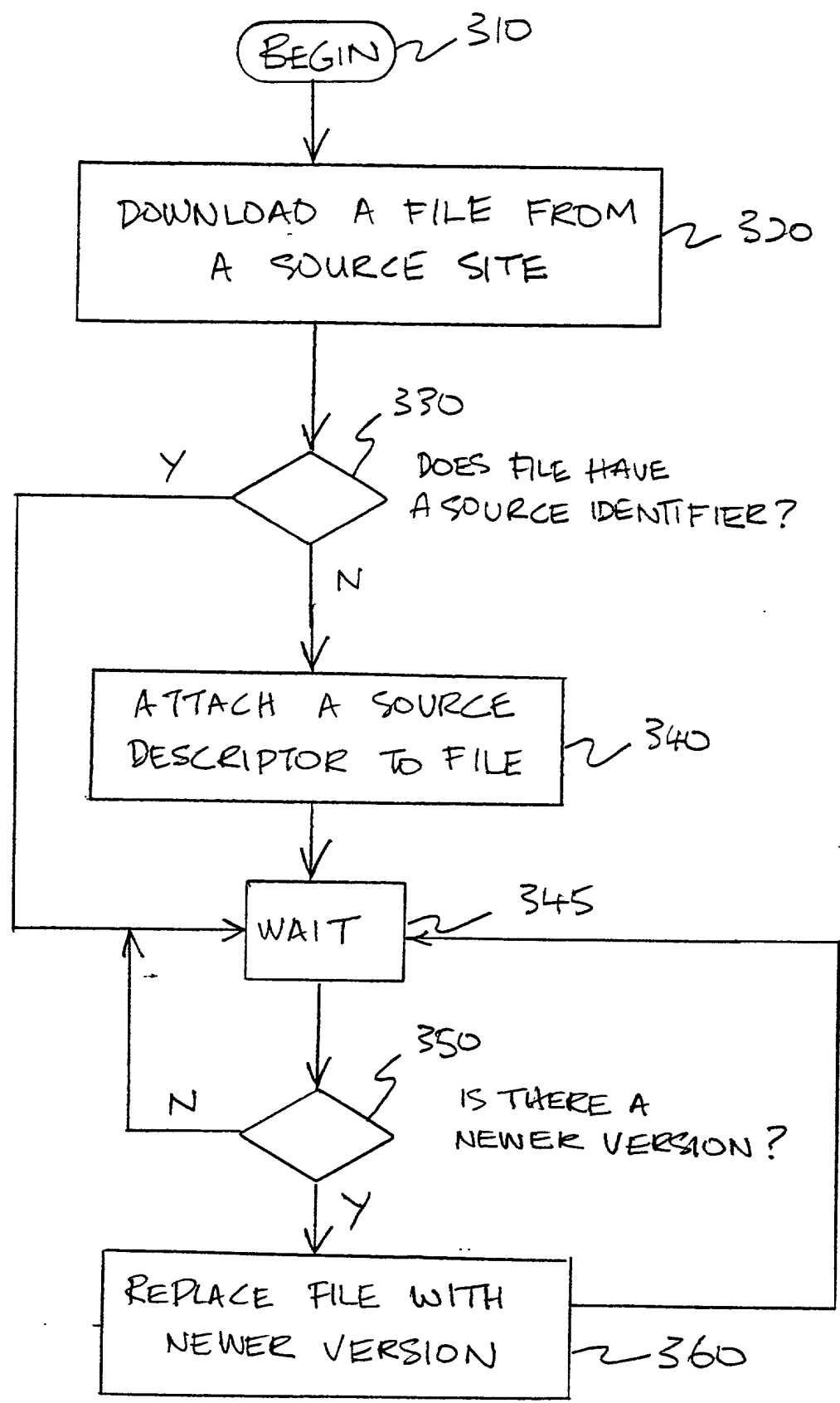NEWER VERSION?

Y

REPLACE FILE WITH
NEWER VERSION 360

FIGURE 3

DECLARATION AND POWER OF ATTORNEY FOR

PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR KEEPING FILES CURRENT

the specification of which (check one)

_X_ is attached hereto.

___ was filed on _____
as Application Serial No. _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):   Priority Claimed

_____ _____ _____   ___ Yes___ No
   (Number)   (Country)   (Day/Month/Year)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

_____ _____ _____
(Application Serial #)   (Filing Date)   (Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; Thomas E. Tyson, Reg. No. 28,543; Robert M. Carwell, Reg. No. 28,499; Jeffrey S. LaBaw, Reg. No. 31,633; Douglas H. Lefeve, Reg. No. 26,193; Casimer K. Salys, Reg. No. 28,900; David A. Mims, Jr., Reg. No. 32,708; Richard A. Henkler, Reg. No. 39,220; Volel Emile, Reg. No. 39,969; James H. Barksdale, Jr. Reg. No. 24,091; Anthony V. England, Reg. No. 35,129; Leslie A. Van Leeuwen, Reg. No. 42,196; Marilyn S. Dawkins, Reg. No. 31,140; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Matthew S. Anderson, Reg. No. 39,093; Matthew W. Baca, Reg. No. 42,277; Michael R. Barre, Reg. No. 44,023; Max Ciccarelli, Reg. No. 39,454; Andrew J. Dillon, Reg. No. 29,634; Justin M. Dillon, Reg. No. 42,486; John G. Graham, Reg. No. 19,563; Andrew H. Harris, Reg. No. 42,638; Steven Lin, Reg. No. 35,250; Richard N. McCain, Reg. No. 43,785; Jack V. Musgrove, Reg. No. 31,986; Antony P. Ng, Reg. No. 43,427; Brian F. Russell, Reg. No. 40,796; and Danier Venglarik, Reg. No. 39,409.

Send correspondence to: Andrew J. Dillon, FELSMAN, BRADLEY, VADEN, GUNTER & DILLON, LLP, Suite 350 Lakewood on the Park, 7600B North Capital of Texas Highway, Austin, Texas 78731, and direct all telephone calls to Andrew J. Dillon, (512) 343-6116.

FULL NAME OF SOLE OR FIRST INVENTOR: Viktors Berstis

INVENTORS SIGNATURE:_____ DATE:__9-13-1999__

RESIDENCE: 5104 Cuesta Verde
Austin, Texas 78746

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: 5104 Cuesta Verde
Austin, Texas 78746